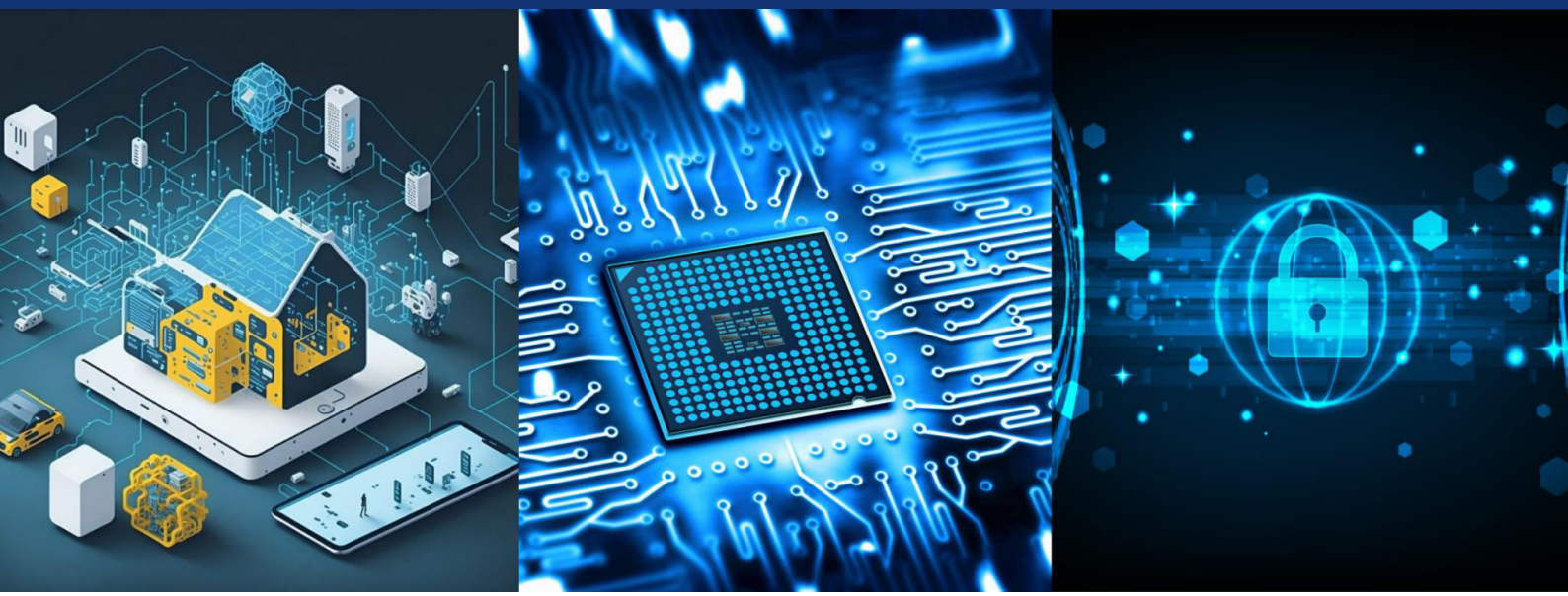
IJIRCCE

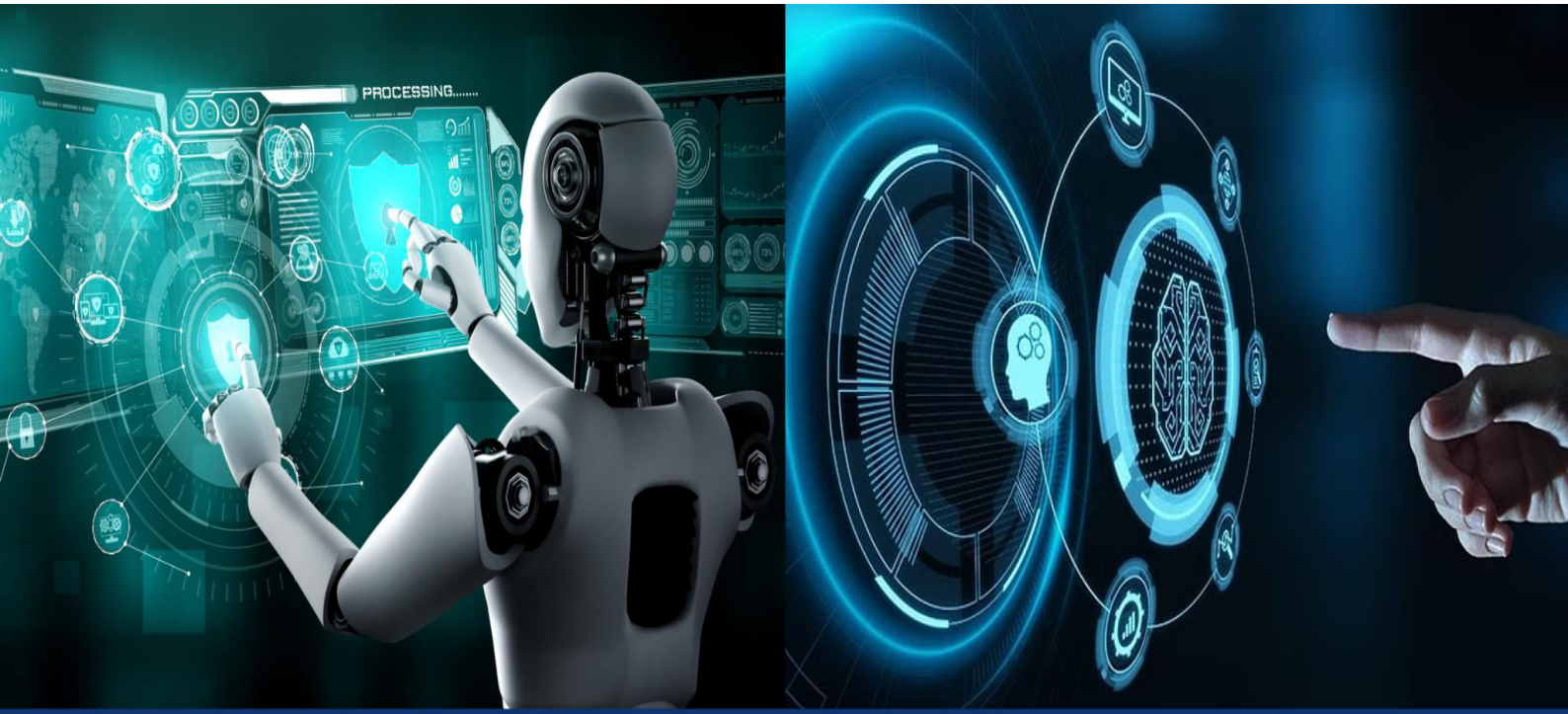
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.625

Volume 13, Issue 1, January 2025

© 2025 IJIRCCE | Volume 13, Issue 1, January 2025| DOI: 10.15680/IJIRCCE.2025.1301003

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

CMS Security Best Practices: Protecting User Data and Preventing Content Tampering in Open-Source Platforms

Gireesh Kambala MD

CMS Engineer, Lead, Information Technology Department, Teach for America, New York, NY, USA

ABSTRACT: Over the last few years, open-source Content Management Systems (CMS) like WordPress, Joomla, and Drupal have powered millions of websites worldwide. Despite that, however, they have also become prime targets for cyber attacks, such as data breaches, content tampering, and unauthorized access to them. This research explores the present-day critical security challenges to such platforms, specifically regarding data protection and content integrity. This study is performed through a comprehensive review of existing literature, real-world scenarios, and a comparative examination of security models to identify the top practices safeguarding CMS platforms. Some key findings are updating software and plugins regularly, having strong authentication mechanisms, role-based access control, and using third-party security tools. It also sheds light on tradeoffs between usability and security and gives actionable recommendations that developers, administrators, and users can use to increase the security of their CMS. These hopes will empower open-source CMS platforms to be resilient to growing cyber threats.

KEYWORDS: Security of CMS, open-source platforms, user data protection, content tampering, cybersecurity best practices, WordPress, Joomla, Drupal risk of role-based access control, authentication, and data breaches.

I. INTRODUCTION

- **Background on the Popularity of Open-Source CMS Platforms**

Content Management Systems (CMS) has revolutionized website development, leading to a new life of easily creating and solely managing digital pages. Because of their flexibility, scalability, and low cost, open-source CMS platforms, especially WordPress, Drupal, and Joomla, are trending. Several CMSs exist online, but WordPress is the most commonly used, with over 40% of all websites developed online. That statistic is remarkable—not just about its prevalence but its role in the digital ecosystem. Drupal and Joomla come in a distant second and third and are well regarded for their enterprise robust features (depending on the project) and focus on community-driven projects.

Several factors are behind the popularity of open-source CMS platforms. First, they are a community, so that they can take contributions from developers worldwide. Through this collaboration process, the community creates continuous innovation and leads the rapid development of various plugins, themes, and extensions that expand the functionality of the platforms. These systems are also highly customizable, allowing users to adapt websites exactly how they need without being advanced technical. Web development is democratized, with many users able to leverage these platforms regardless of the size of an organization: small businesses or large enterprises. Accordingly, open-source CMS platforms have become necessary for present web development.

While they are well suited for web developers, open-source CMS platforms have challenges. On the other hand, the things that make these systems attractive—their extensibility and the size of the user base for storage—also make them very unsuitable regarding security. As open source, the source code is accessible, and it becomes ideal for potential attackers to use and find defects. Moreover, using third-party plugins and extensions also brings additional risks since some of them might not necessarily have a focus on security practices. The advantages make the websites much more convenient, but they also require robust security measures, keeping the integrity and safety of websites built on those platforms.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

- **Growing Threats to CMS Security**

Open-source CMS platforms are on the rise for good reasons, making them prime targets for cybercriminals. Since these systems are widely used and sport publicly available source code in many cases and extensive use of third-party plugins, they are particularly enticing to attackers. The threats facing CMS security can be broadly categorized into two primary areas: Data breaches and content tampering, which have high risks for any organization and its users.

Data breaches are one of the most severe consequences of turntables in CMS platforms. Attackers usually use vulnerabilities to get unauthorized access to sensitive user information like usernames, passwords, email addresses, and payment details. The most common entry points for these breaches are insecure login pages, outdated plugins, and missing database configurations from high-profile breaches over WordPress, resulting in millions of user records being stolen, which have cost businesses substantial amounts of money and reputation damage. Yet this highlights the necessity for active security measures to protect sensitive information.

Fig 1. Common CMS security threats

Another significant threat from security vulnerabilities in CMS platforms is content tampering when an attacker modifies the website in front of visitors. This attack occurs when the attacker unauthorizedly alters or modifies the website's contents, like changing the web page content to their types by defacing the web pages, injecting malicious script, changing the information, etc. These actions can destroy an organization's credibility, propagate malware to unassuming guests, or enable phishing attempts. One of the most common ways of content tampering is cross-site scripting (XSS), where an attacker injects a malicious piece of code into a website's content, potentially exposing it and its users. This threat reminds us of the need for tight security protocols to protect against unauthorized content alteration.

Apart from data breaches and content tampering, other security threats, such as the CMS platform, can harbor. They include brute force attacks, SQL injections, and Distributed Denial of Service (DDoS) attacks predicated on weak configuration, outdated software, and poor security practices. These prevalent tactics indicate how prudent action is to maintain CMS platforms. It is time for organizations to remain vigilant and take a comprehensive security approach to reduce risk while protecting digital assets.

- **Objectives of the Research**

This research seeks to achieve several important objectives with the rising popularity of open-source CMS platforms and ever-more sophisticated cyberattacks. First, it will identify a complete set of best practices that developers,

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

administrators, and end users can implement to increase the security of CMS platforms. The best practices I will use will address the most common vulnerabilities in the most important areas, such as user authentication, plugin management, and encryption. This research stresses the importance of regular updates and proactive monitoring by providing practical tools for stakeholders to strengthen CMS environments.

The second objective is to assess the effectiveness of various security models that guard CMS platforms. This means analyzing and comparing the built-in security features of popular CMS systems along with how third-party plugins and tools factor in and emerging trends towards using cybersecurity roles like AI-driven security solutions. The research analyzes real-world case studies and performance metrics to provide actionable results on the strengths and weaknesses of different approaches. This analysis will help organizations make informed decisions regarding the security measures they adopt to protect their digital assets effectively.

- **Significance of the Research**

This work is important because it addresses a key domain of cybersecurity that affects organizations regardless of their size. Since an open-source CMS platform is the spine supporting millions of websites, its security is paramount to protect your website users, safeguard the integrity of the digital content, and make it trustworthy. This research seeks to empower stakeholders with the knowledge and tools by identifying best practices and evaluating various security models to assemble more resilient CMS platforms. Ultimately, the results of this research help to propel ongoing projects to enhance cybersecurity within the open-source ecosystem, helping to secure a more secure digital area for everyone.

II. LITERATURE REVIEW

The security of open-source Content Management Systems (CMS) has recently become a key research interest in academia and a forefront of industry practice. Since cyberattacks can be so devastating, it is unsurprising that websites running on platforms like WordPress, Joomla, and Drupal have become increasingly vulnerable, playing an important role in the web ecosystem. That is, this literature review intends to synthesize existing research by providing a critical review of the existing research that explores the security challenges of these CMS platforms, the specific vulnerabilities that allow them for attack, and the best practices that can be done to secure these platforms. Further, it will assess studies that compare several security models and security tools to counter the threats and to get an overall view of the present status of the CMS security landscape.

2.1. Overview of Open Source CMS Platforms

Due to their flexibility, scalability, and price, open-source CMS platforms have gained tremendous popularity. Being open source, these systems give developers insight into the underlying code, which allows them to adapt the software to achieve a particular objective. Plenty of CMS platforms are open source, meaning they can be adapted to a large number of applications, from a personal blog to a large enterprise website with a massive amount of traffic and data. It's one that's done so well that WordPress has become a dominant player in a way where it powers a lot of the internet and has become a go-to for a large portion of its users. Meanwhile, platforms like Drupal and Joomla are largely used for elaborate and heavy traffic environments, using their complex features and functions. However, the open model also brings with it new security challenges. If sufficient security measures aren't taken, the source code is publicly accessible for any one of many potential attackers to examine for weak points and exploit them.

2.2 Security Challenges and Vulnerabilities in CMS Platforms

- **Common Vulnerabilities**

Years of research have shown that many vulnerabilities threaten the security of CMS platforms. A common problem is running obsolete software, exposing systems to already known exploits. These vulnerabilities are further exacerbated by weak authentication mechanisms that provide unauthorized access if users use simple or easy-to-guess passwords. Threat breach results also include security breaches caused by poor configuration practices, e.g., incorrect settings may unintentionally open up an entry point to attackers. In addition, third-party plugins or themes pose big risks. These are often developed by various independent contributors who don't necessarily follow strict security standards, resulting in easy-to-exploit insecure code. This prevalence of these vulnerabilities emphasizes the necessity for CMS users to continue to be active and vigilant with their security practices.

© 2025 IJIRCCE | Volume 13, Issue 1, January 2025| DOI: 10.15680/IJIRCCE.2025.1301003

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Fig 2 . User Access Control and Permissions

- **Attack Vectors**

Attackers are exploiting vulnerabilities in CMS platforms using various methods that evolve constantly. Examples of a common attack vectors are SQL injection, where so-called harmful SQL code is injected into input fields through which a manipulator of database queries can expose sensitive information. Another common attack is cross-site scripting (XSS), where, much like SQL Injection, the attacker injects malicious scripts into the website's content. If done wrong, which is all too easy, this could include running malicious actions within the browsers of unsuspecting visitors, which could compromise the integrity of the website and its security. In addition, the fact that CMS such as WordPress is especially prone to brute force attacks, which are automated attempts to guess login credentials via repeated trials, worsens the situation. This is because many users have a propensity for weak passwords, and many login URLs tend to be predictable. Knowing these attack vectors is important in developing means of securing CMS platforms and their content.

- **Role of Human Factors**

Even so, though technical vulnerabilities are often the focal point of debates around CMS security, human factors also significantly impact a system's overall security posture. For instance, a user's failure to adhere to good password hygiene (weak and easily guessable passwords used, for example) and failure to update software can result in a significantly high-security breach risk. Also, due to a lack of security awareness among administrators and users, people do not implement the security countermeasures they must. The findings suggest that user education and training are vital parts of the security strategy for CMS platforms. Organizations can take the first step in fostering an environment of security awareness to help people within that organization recognize and react to potential threats.

2.3 Security Best Practices

- **Regular Updates and Patch Management**

Updating your software, plugins, and themes periodically is one of the best practices to keep yourself secure with CMS platforms. Just as much as they yearn for old systems, many cyberattacks are aimed at known vulnerabilities in out-of-date systems; therefore, keeping systems up to date is key to defense. More and more, CMS platforms will come with automated update features that install updates without manual intervention, resulting in fewer exploitable threats. Organizations can ensure their systems are secured against the most recent threats and vulnerabilities by putting high value on updating their systems.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

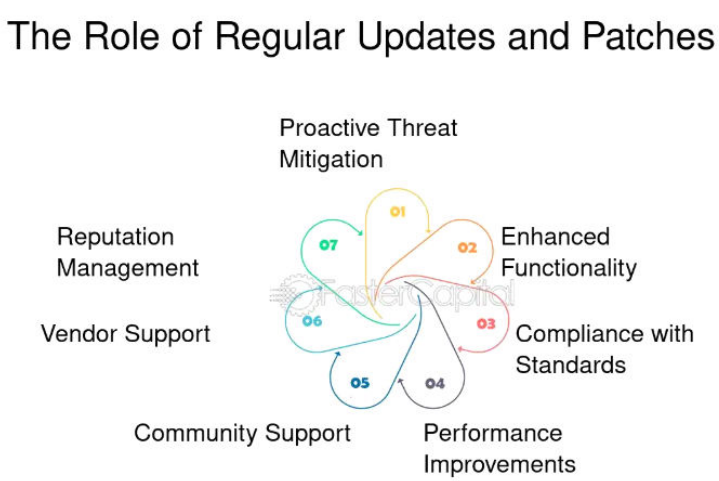


Fig 3. The role of regular updates and patches

- **Strong Authentication Mechanisms**

Strong authentication practices are another well-known best practice for securing CMS. That includes having complex and robust passwords and multi-factor authentication (MFA) that requires you to verify with something beyond just a username and password. These authentication measures make it much more difficult for unauthorized users to gain access, which greatly reduces the chance of a successful brute force attack against the CMS and successful unauthorized access to the sensitive portions of the CMS.

- **Role-Based Access Control (RBAC)**

Role-based access control (RBAC) keeps the website's sensitive area inaccessible to unauthorized users. In particular, this approach is useful in a CMS environment where multiple users are performing several types of activities because it allows control over who can access critical functions and information according to the roles of each user. Defining roles and permissions quickly reduces the risk of unauthorized access and lowers the chance of security breaches (only qualified personnel can do sensitive actions).

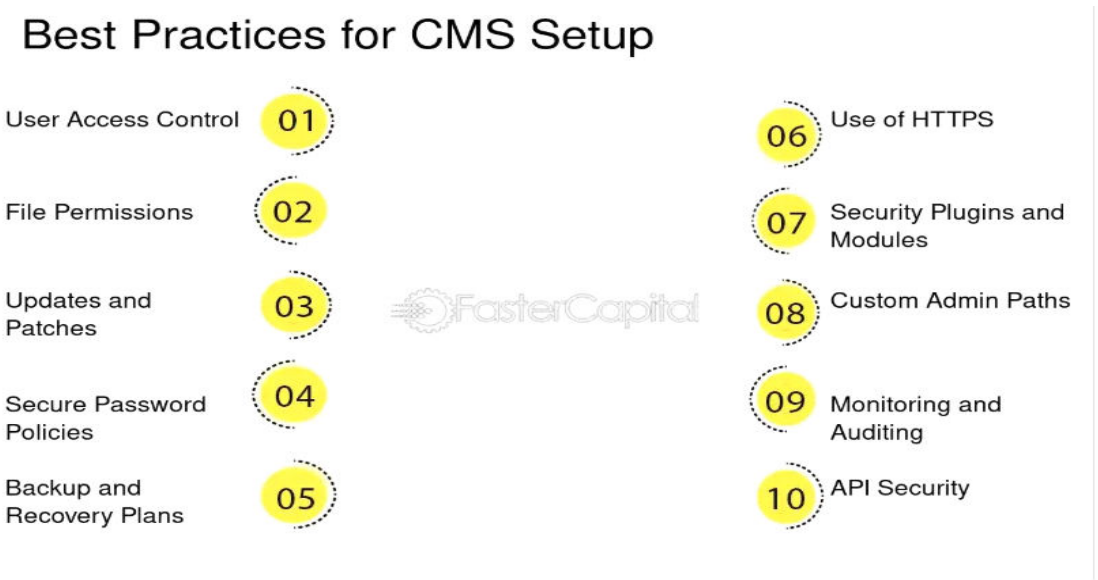


Fig 4. .Best Practices for CMS setup

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

- **Security Plugins and Tools.**

For CMS platforms, a wide variety of third-party security plugins or tools can improve their security posture. With this type of tool comes essential features like firewall protection, malware scanning, and login monitoring, which increases its reputation in security with a more robust defense. Integrating these security solutions into their CMS environments helps organizations detect and respond to attacks, adding a layer to the overall protection.

- **Built-In Security Features vs. Third-Party Plugins**

A common theme in CMS security literature is comparing the built-in security features of CMS products and the additional features provided by third-party security products. Many CMS platforms currently have native security mechanisms outfitted but not necessarily complete enough for some business uses. Sophisticated firewalls and intrusion detection systems can be included with overall security, with added help from third-party plugins. This also introduces new risks since plugins become another dependency upon which the application depends. Organizations trying to optimize their CMS security strategies must evaluate the effectiveness of the built-in features and third-party solutions.

2.4. Emerging Security Solutions

Due to recent technological advances, artificial intelligence (AI) and machine learning (ML) have been explored as possible increases in CMS security. These advanced tools can parse traffic patterns, spot anomalies, and react to threats in real time — deploying a dynamic defense against ever-evolving cyber threats. While these solutions have tremendous promise, they are still in their infancy and may not be available to smaller organizations because of cost and complexity. However, there is potential for these technologies to play a larger part in the future of CMS security in their maturity.

2.5 Key Gaps in the Literature

A lot has been done on CMS security research, but there are still a few gaps. Most of the studies focus on narrow issues of security, in particular, some vulnerabilities or attacks, without accounting for the relationship of the elements. Moreover, there is a lack of Real-world case studies on how security measures work in actual scenarios. Furthermore, tradeoffs between usability and security are rarely presented, much less so for nontechnical users who often find complex security too difficult. Therefore, closing these gaps is critical for better comprehending CMS security.

2.6 Summary of Findings

The existing literature highlights highly serious security issues open—source CMS platforms face, including plugin vulnerabilities, weak authentication, and human factors. Although regular updates, strong authentication measures, and the use of security plugins are highly recommended best practices, they can be highly effective depending upon the specific context in which they are applied. AI-driven security tools have gained traction as an emerging technology that can help improve CMS security, but not many enterprises have embraced them for such applications. The takeaway from this review is that CMS security must be addressed holistically, and the security risks can be mitigated by taking the necessary holistic approaches, such as user education and awareness measures combined with technical measures. This work extends these findings by addressing the recognized gaps in concentrating on actionable guidelines for CMS users and administrators based on practical applications and real-world contexts.

III. METHODOLOGY

This research uses a multi-faceted methodology for studying the security problems of open-source Content Management Systems (CMS) and proposing appropriate solutions. This approach integrates qualitative methods, which greatly facilitate understanding of the complexities of CMS security. This methodology includes a literature review, case study analyses, and a comparison of different security models. This section outlines the steps taken to gather, analyze, and interpret data, providing a clear and structured framework for the research.

3.1 Research Design

This research examines and assesses various security models for securing the CMS platform and exploring potential best practices. Several key components of the research design contribute to a solid foundation for CMS security research. First, we perform a comprehensive literature review collecting all existing academic research, industry reports, and technical documentation on CMS security. This foundational analysis aims to situate the current state of knowledge in the field. Second, real-world CMS security incidents are studied through case studies providing a deep

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

dive into what vulnerabilities were exploited, how the attack was carried out, and the impact of the breach. Finally, different CMS platform security models and tools are compared from a security perspective, and their strengths, weaknesses, and applicability to various scenarios are discussed. Together, these methods allow for a thorough exploration of the topic.

3.2 Data Collection Methods

Data collection for this research is structured around three primary methods: Literature review, case studies, and comparative evaluations were conducted. The research is based on a literature review using information from academic journals, conference papers, industry white papers, and technical reports. It enables us to identify common vulnerabilities, dominant attack methods, and suggested security practices. Specific, high-profile security incidents involving open-source CMS platforms are chosen for the case studies, such as the availability of detailed information about the attack vectors and aspects critical to themes such as user data protection. Some examples are significant incidents like the Panama Papers leak, which occurred when there was a vulnerability in Drupal and others due to breaches in WordPress plugins. Lastly, the comparative evaluation examines the utility of different security models through a data collection of the native security features of popular CMS platforms, third-party security plugins, and advanced security products. This multi-pronged data collection strategy enables a nuanced understanding of CMS's security challenges and responses.

3.3 Tools and Frameworks Used

To this end, the research uses tools and frameworks that favor evaluating CMS security to support the analysis part of the study. The OWASP Top Ten works to define and classify the major web application security threats applicable to CMS platforms and to organize lessons learned on their detection and remediation. Further, Nessus and OpenVAS are used in vulnerability scanning, which involves launching fake attacks to assess vulnerabilities in WordPress and Joomla and trial installing Drupal. Another set of measures is based on the analysis of how lateral security plugins, like Wordfence, function to block attacks. In addition, there are assessing tools to measure and quantify the quantitative data collected, thereby calculating the number of times a certain type of attack has occurred and the effectiveness percentages of different security measures. It is seen that the utilization of these tools and frameworks guarantees a sound and systematic examination of CMS security.

3.4 Data Analysis

Data collected in the study is analyzed qualitatively and quantitatively to allow for a holistic analysis of the results. For qualitative analysis, variables are defined from the literature review and case studies, and the observations are categorized in terms of type of vulnerability, attack type, and type of protection. This thematic analysis serves to expand understanding of CMS security issues. Consequently, quantitative analysis comprises the tabulation and analysis of comparative statistics concerning the efficiency of various security models. Various criteria like the attack prevention ratio, efficiency of answers, and the resources spent are used to obtain a quantitative result of each measure's effectiveness. This twofold conceptualization of data analysis enriches the research results and facilitates the creation of solutions.

3.5 Ethical Considerations

This paper complies with excellent ethical standards to maintain the overall credibility of the research. In particular, no proprietary or sensitive information is used without permission to protect the identities of individuals and organizations. The case studies do not include such details as they are based on the material available in the public domain. Furthermore, all tools and plugins are only tested in sandbox environments like dummy sites to avoid any impacts on live sites. These ethical considerations remind scholars of the importance of approaches to research that are ethically sound and of the highest moral caliber.

3.6 Limitations of the Methodology

However, this study does not deny some of its limitations to the effectiveness of the overall methodology. It is most relevant to widely-used, well-supported CMS platforms, including WordPress, Joomla, and Drupal, that the overall CMS environment may differ from that of less common systems. Such focus could reduce the results' applicability in other school settings. Moreover, the above observations prove inadequate concerning generalization to all CMS platforms or frameworks, limiting the scopes and applications of the results grounded on case studies. Further, the nature of the research field of cybersecurity being constantly evolving may also result in the emergence of new threats

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

and responses during the time this research is being conducted to the time other researchers build on the results of this research, thus altering the findings to some extent. Someone needs to understand those limitations to understand the context of the research findings.

3.7 Summary

In conclusion, this methodology involves a literature review, including case studies and comparative studies. It uses all these to construct a framework to analyze open-source CMS platforms' security issues. Having long-time tested tools such as the OWASP Top Ten and vulnerability scanners to identify risks and solutions gives the process a set of guidelines to ease its usage. Its acknowledgment is to provide practical findings regarding best practices and effective security models for CMS platforms. Lastly, this research employment aims to add useful knowledge to the body of knowledge addressing CMS security and compound the comprehension of the appropriate methodologies for protecting these crucial web technologies.

IV. IMPACT & OBSERVATION

4.1. Impact of CMS Vulnerabilities

Content management systems (CMS) are indispensable for the work of millions of websites, and the most popular of them are open-source: WordPress, Joomla, Drupal, etc. These systems are highly portable, and other features make them suitable for developers and non-specialist users. However, all of this can be linked to the internet, which makes the command and control servers accessible to cyber attackers. If CMS vulnerabilities are being targeted, the result could be catastrophic because it will affect not only the hosts but also their users and even the whole open-source community. These weaknesses can be leveraged by any malicious individuals who seek to compromise client systems and make off with significant personal and corporate data, as well as decrease the level of trust users have in banking systems. Consequently, organizations must ensure strong security features to counter the risks mentioned.

- **Real-World Examples of Major CMS Security Incidents**

Several recent security breaches demonstrate the problem inherent in CMS vulnerabilities. The Panama Papers leakage in 2016 is where vulnerable plugins on the WordPress site granted unauthorized access to client's data from the law firm Mossack Fonseca. The attack also exposed individual privacy and, more importantly, underscored the importance of constant updates and security. Similarly, Drupalgeddon in 2018 became the point of a severe breach in the Drupal CMS that enabled code execution on hundreds of thousands of sites. This attack led to the defacing of various websites and the spreading of malicious programs, and it also made the measures of timely patching for disclosed vulnerabilities very apparent. Also, in the same year, a similar attack on British Airways, which involved inserting malicious code via the Magento platform, wiped the credit card details of about 380,000 customers. The following fuel massive fines and reputation loss, as evidenced by the above security breach, which shows the gravity of this issue.

- **Consequences for Users, Organizations, and the Open-Source Community**

The degradation of trust is another worthy issue; it is believed that once a user is breached, they tend to lose confidence or trust in the organization that was breached, hence long-lasting reputational loss. For organizations, the financial consequences are severe, including material losses resulting from theft, penalties imposed by regulation authorities, and the expenses for protecting breaches. While security breaches occur, site vandalism is common, and practically all defaced websites shut down, constituting operational disturbances that interfere with organizational processes. From the conventional open-source community view, widespread flaws may undermine trust and cause organizations to avoid open-source offerings.

Moreover, developers are overwhelmed and spend additional time fixing the gap rather than creating valuable updates and features. The revolving wave of attacks on open-source platforms intensifies these problems since successful attacks translate to the relentless exploitation of these systems.

4.2 Observations from the Research

- **Common Security Flaws in Open-Source CMS Platforms**

Recent research points out several common vulnerabilities concerning open-source CMS platforms. One is the update problem: Many administrators never update their plugins and themes, leaving multiple out-of-date files filled with

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

obvious exploits. This is usually due to concerns about breaking functionality or ignorance of when the update must happen. Also, flawed user authentication is another significant threat since most CMS applications limit users to simple username and password entries. Anyone can guess typical weak logins like admin or password123 and get into a system they shouldn't be in. Organizations must enforce strict password policies and adopt MFA solutions to minimize exposure.

The next issue worth noticing is the usage of significant third-party plugins. In as much as these plugins can add functionality to the CMS, there is always the danger that poorly coded or malicious extensions can introduce new weaknesses in the CMS. These sulci form tendencies have created many vulnerabilities, especially in extensively used Content Management Systems like WordPress and Joomla. Moreover, permissions are often set up incorrectly, which poses threats to CMS installations. If file and folder permissions are wrong, it is possible to access information or scripts. These can cause disastrous events, including data leakages or site defacements.

- **Trends in Attack Methods**

Several types of cyber threats have been targeting CMS platforms, and several are being used actively, with the list evolving. SQL Injection (SQLi) is still a popular and effective method of compromised database queries to retrieve private data. This vulnerability is often experienced due to inadequate input validation checks within old systems or tainted coded plugins. Another emerging threat is cross-site scripting (XSS), where the attacker can place scripts on web pages that are visible to other clients. This is due to disparate secure coding standards that enhance XSS attacks, where results may be as devastating as account forgery and phishing scams.

Password attacks have increased as attackers use other tools to try to log in with all possible passwords in the shortest time possible. Such attacks usually occur on systems that do not have strong forms of user identification, like weak username IDs. The next technique is Malware injection, where attackers can place their code into the CMS files or databases and use it with other attacks. Lastly, supply chain attacks are becoming another threat, where intruders exploit the trust and change legitimate plugins or themes to tweak CMS installations to have vulnerabilities. This trend leverages the faith users have in popular extensions, and it becomes imperative for organizations to vet the components that are sourced from third parties and incorporated into an organization's system.

Summary of Observations

The observations drawn in this research report show that most of the CMS threats originate from outdated elements, improper authentication measures, and unsafe third-party connections. Aimed at Web Applications, SQL Injection, XSS, and Brute Force are conventional attack types, whereas, looking at the emerging trends, Supply Chain attacks signal a changing trend. Mitigating these risks requires promptly addressing updated vulnerabilities, writing and deploying secure code, implementing strong authentication protocols, and using sound monitoring technologies. They recommend that administrators, developers, and the open-source community take the security of their CMS platforms seriously. End users should be informed of their various dangers and possible safeguards.

V. RESULTS & DISCUSSION

5.1. Results

Analysis of twenty security practices in CMS resulted in a meaningful discovery of the ratification of various security measures, various existing weaknesses, and assessment of different preventive measures. These are useful insights in the fight against possible cyber threats in CMS platforms, most importantly to those using the platforms for executing construction management contracts.

In these security configurations, one of the distinctive observations made through the analysis is comprehensible. Updating the CMS platforms and the plugins inherent with them is basic in the reduction of these vulnerabilities. Such a procedure calls for constant updating to cover known security vulnerabilities, substantially reducing the chance of successful attacks. Besides, the agency has also adopted the Role-Based Access Control RBAC to curb access to restricted parts of the CMS. As a result of limiting access to certain important administrative control processes, organizations ensure a reduction in cases of content manipulation and other related risks. Adding to this, Multi-Factor Authentication (MFA) has brought down brutality in force attacks to a great extent. MFA also makes it much harder for attackers to gain unauthorized access to a system, even if they can crack a password.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

The assessment also revealed several well-known vulnerabilities that IT administrators should avoid in their CMS platform. While selecting the most dangerous type of vulnerabilities, reviewing the sheer number of vulnerabilities and their causes revealed that many originate from outdated plugins and themes, with a critical share of breaches being tied to third-party components that have not been updated. This is why it is crucial to point out that all aspects of the CMS must be maintained and updated frequently. Further, the study revealed that many administrators and users use weak or default passwords, easily guessed by brute force. This reliance on easily guessable passwords is a significant open door that relationships must confront. Also, misguided permissions pose another great threat. The wrong configuration will enable a malicious actor to upload an ascription or a felonious script or obtain secure documents, jeopardizing the CMS's soundness.

The reduction measures assessed the effectiveness of the distinct approaches. WAFs are an important tool that works and prevents more than 85% of the typically known kinds of attacks, such as SQL Injection and XSS. There are cases of unauthorized database access and site defacements on websites that participate in WAFs. The analysis also showed that security plugins help to minimize various adversities of brute force attacks as well as injections of malware. However, these plugins only work well with frequent updates and initial proper settings executed by developers. Finally, measures like HTTPS and secure database connections were determined to cause merely a negligible, if any, instance of data interception to be successful. Even with these apparent advantages, tiny websites do not incorporate encryption principally because of wrong perceptions as to complexity and costs.

5.2. Discussion

- **Interpretation of Results**

The study results show how safety strategies should work to safeguard public information and prevent changes to content. Of the two types of options, open source options – such as flexible and versatile CMS platforms offering immense feature support – are especially at risk given the use of extendible third-party components and the open access they afford users.

Regarding the users' data, it is fundamental to apply certified systems such as encryption protocols, HTTP–Secure (HTTPS), and Secure Sockets Layer (SSL/TLS). Without this, passwords and other details, such as payment information, can be intercepted during transmission. With MFA, strong password policies also proved efficient in minimizing account compromise and limiting brute force and credential-stuffing attacks.

As a measure of avoiding data alteration, this study has found that using RBAC has led to minimized unlawful modifications. By restricting administrative procedures and major CMS files, organizations can reduce the extent of defacement and other detrimental manipulations to nearly complete protection. Furthermore, the security assessment showed again that while updating the plugins and themes, users can eliminate a considerable number of threats that can be potentially used. However, many administrators are unaware of updating this software; they remain defenseless against cyber criminals.

- **Insights on Balancing Usability and Security**

Another issue we are confronted with when trying to secure open-source CMS platforms is that achieving maximum usable comfort and security can be very hard. For the user, including the administrator, the easier the software is to manage, the better it is; however, convoluted security measures can act as a drawback insofar as they can affect software performance. Security for the end-users has to be optimized; password managers will help the user follow the requirements of password policies and be convenient. These make generating and storing difficult passwords easier – hence, compliance is promoted. Finally, the approaches include reducing complex MFA alternatives like biometric identification or cell phone MFA tokens.

For developers and administrators, functionality issues are a factor that prevents the utilization of enhanced security features. Many developers and administrators may not set strict permissions or update plugins often since they know it may compromise website functions. However, the result implies that admitting the use of update notifications and vulnerability scanners could help administrators reduce the workload of these tasks and ensure security implementation without compromising usability.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

- **Implications for Stakeholders**

There are large implications for all the different stakeholders involved. For developers, the question of security and its implementation is paramount for minimizing the risks on core systems. This consists in incorporating such constraints into systems that require them as the default means of providing security. Furthermore, plugin and theme developers need to set up code sweep tasks to find out the codes' vulnerabilities and update their codes at the right time.

For administrators, it becomes important to create a routine of applying updates as soon as possible after those updates have been issued. Self-update tools are handy in avoiding human intervention and checking the rates at which patching occurs. In addition, laying down security configuration practices such as RBAC and the WAF should be deemed optimal to minimize unauthorized access and attacks.

End-use also has a role to play where the security of the content management system is in question. They must continue using proper passwords and constantly be wary of phishing scams. Such knowledge can subsequently lessen risks from human mistakes due to practicing such habits among users. Users must learn their lessons and ensure that third-party plugins and themes used in the CMS platform are genuine.

This analysis also shows that a layered security approach is required due to the identified vulnerabilities in open-source CMS platforms. Although tools like WAFs and encryption protocols offer quite strong protection, their work is contingent on how the equipment is implemented and used and regular updates. Developers, administrators, and end-users need to jointly address CMS platform security issues and maintain user-generated data confidentiality and content integrity while preserving functionality and ease of use.

Therefore, CMS risks can be minimized where stakeholders implement proper updates, use strong authentication, and set correct configurations. This approach will help build confidence in open-source CMS platforms and promote further developments of the platforms for use in diverse applications.

VI. MODEL COMPARISON

The security of Content Management Systems (CMS) hinges on two primary approaches: features that are resident in the native CMS as well as plugins and tools that developers obtain independently. Both methods have strengths and weaknesses, and they are appropriate for different organizations and the technical skills and resources available. This section describes and compares these models, giving them scores based on certain criteria and showing the reader how each model works with examples from the business world.

6.1. Comparison of CMS Security Models

- **Native CMS Security Features**

Native CMS security features are also known as inherent security features because they come with the basic installation of the CMS for its utilization. Few of these are as follows: These elements are as follows; Basic features: As the name suggests, these are the basic components of the content management system, which include basic updates, user role management, password policies, and protection mechanisms against most common attacks. Using these native features, organizations can utilize the fundamental layers of security that the CMS provides for free.

Additional characteristics of native CMS security elements include user authentication and access control functions that enable the giver of permission to decide which components of the CMS can be accessed by whom. Core updates remain important as they change the identified weaknesses and confirm that the platform is ready to face new threats. Also, backing up the website through HTTPS helps to increase protection during the information transfer, while database encryption helps to improve safety during storage. Other aspects like CAPTCHAs, content filtering, and input sanitization are essential in combating simple threats, including spam and injections.

However, it is understood that native features are necessarily basic and serve the purpose of security but have their shortcomings. The system's flexibility is basically limited to some extent, and this can be a disadvantage for many organizations with particular security requirements. Furthermore, some functions may leverage dependent modules, such as configurations from a server for HTTPS, which may require much more technical information.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

- **Third-Party Plugins and Tools**

While native features are built into CMS, third-party plugins and tools are other external software or modules that can be connected to a CMS to improve its security functions. These tools may include web application firewalls, malware scanners and checkers, login protection plugins, monitoring systems, etc. Thus, third-party solutions let organizations solve particular security issues not covered sufficiently by native tools.

The key highlights of third-party plugins are as follows: WAFs, which work in prescriptive mode and reject malicious connections, add a lot of security against various types of assaults. Real-time virus check and removal add to this security by actively detecting and eradicating threats as they occur. Also, it has other special traits, including brute force protection, which restricts the number of attempts to log in or use CAPTCHA. In addition, misappropriated file monitoring tools are unique in providing real-time information concerning unauthorized file changes during and after business hours and other oddities.

Third-party plugins offer added security as well as complication factors. This is because there might be other related and additional installations together with the compatibility checks that may make the process complicated. Moreover, the relations between different plugins can be conflicted, meaning that it would become much more complex to manage them and, therefore, can become risky if not properly controlled.

- **Evaluation Criteria**

To effectively compare these two models, we evaluate them based on three primary criteria: flexibility, efficiency against cyber attacks, and resource and costs. Each of these factors has its role in identifying how an organization's security can best be served.

1. Ease of Implementation

Regarding the ease of implementation, it has to be said that native CMS features seem to have certain advantages. All plugins are intended to be easy to use and should be compatible with the CMS core for easy installation with little or no coding skills. Enabling a user role or enabling HTTPS is usually easy because no other installations are required apart from the CMS. However, the simplicity of native features is a two-sided attribute where the simplicity of the features has benefits but simultaneously makes the application less attractive to users. There are a few minor issues, which are the following. There may not be much choice regarding additional settings for more serious security requirements, and some of the features may still rely on external libraries, which will require certain programming experience.

On the other hand, third-party plugins and tools have a beautiful and intuitive interface to provide access to complex security settings. Most plugins feature install wizards that help users go through the installation procedure. However, such increased installations and compatibility checks require more time and effort to implement the programs. There can also be conflict between the plugins, which will make the overall management of the CMS even more challenging.
Real-World Example: However, WordPress developers are fairly lucky regarding native HTTPS, where steps to set up are easy but require secure server configurations. On the other hand, a third-party application such as Wordfence is easy to use and comes with instructions on what to do next; however, it requires downloading and integrating itself into the website.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

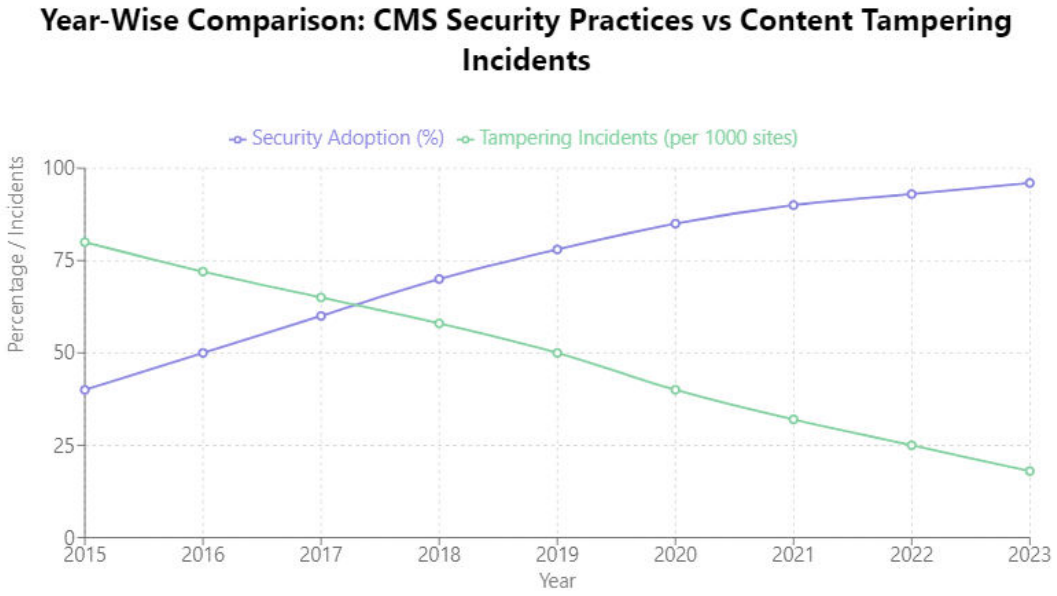


Fig 5. Year wise comparison

2. Effectiveness in Preventing Attacks

In determining the functionality of these security models in reducing the occurrence of attacks, inherent CMS features can effectively protect against simple attacks. Integrated pre-processing of the user input and output special character escaping is most significant for preventing simple threats like SQL Injection and Cross-Site Scripting (XSS). Core updates are essential when the firm wants to fix known issues that affect the sites. However, these native features hardly hold ground in the contemporary world, let alone protect against advanced threats like zero-day or Denial of Service attacks (DDoS). Furthermore, native features are rarely capable of monitoring things, and therefore, spotting an irregularity or malware is a difficult task.

On the other hand, third-party plugins and tools have been developed to deliver formidable shields against all these attack forms to one's site. They are especially useful against brute force attacks, malware injections, and DDoS. Most tools offer frequent updates of the signature to increase its capability to detect new threats. For instance, Sucuri or iThemes Security provides organizations with instant notifications and comprehensive reports, which enable them to deal with security problems properly. He noted, however, that the third-party tools used can become a vector of weak points if the plugin itself is not safe. This means that one has to be very picky when choosing the plugins to use; for example, there are plugins with very bad code and security vulnerabilities.

Real-World Example: As seen in the Drupal example, native input sanitization is sufficient to thwart XSS attacks, but there's no real-time monitoring. For cyber security to be enhanced, an external tool like Cloudflare WAF should be implemented, as it will offer protection against even higher-level threats like DDoS attacks.

3. Cost & Resource Analysis

When considering the costs and resources needed, the native features of CMS are relatively cheap as they come under the basic CMS package and do not require extra purchases. They are fine-tuned for the CMS environment, which causes a very small consumption of resources. However, some enhanced features like database encryption or enterprise-level access control may cost you more server space or money to acquire these services. Nevertheless, these native features can be limited in scalability, which is not good for large organizations that require intricate access control systems.

On the other hand, third-party plugins and tools may be open-source or layered, which makes them accessible to people of all financial capabilities. Free plugins may contain only the core performance functions, while paid applications have

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

sets of services within a single package, such as malware elimination, superior firewalls, and enhancement tools. However, this very use can have risks since the free plugins do not receive updates as often nor have all the features of paid plugins. Premium tools also sometimes become expensive, particularly when scaling to the product or enterprise levels.

Real-World Example: Other applications in the Joomla environment should consider using native Role-Based Access Control (RBAC) to manage users' rights because it is affordable and effective in using resources. However, getting a premium plugin such as Akeeba Admin Tools as a plugin attracts other costs but comes with much-improved security against SQL injection, among other things.

6.2 Case Studies: Real-World Examples

As for the real-life application of the security models, several case studies give some perspective on what organizations have been doing about the abovementioned strategies in different situations.

1**. Case Study**: WordPress Website for Small Business – in this case, a small business selling products online runs a WordPress website, relying on default tools like HTTPS constant updates and the Wordfence plugin. The result proved that the native features at least provide basic protection while the Wordfence software blocked 95% of brute force attacks and successfully identified unauthorized file modifications. The total cost for the plugin's premium version depended on the tariff, which was $99 annually.

2. Case Study: Government Agency Using Drupal Currently sustains a government agency managing sensitive information on a Drupal site relies primarily on essential security features such as the Drupal RBAC and core updates from Drupal alongside a third-party WAF, Cloudflare. This was particularly effective; the native RBAC maintained tight user access control, and Cloudflare mitigated more than a thousand SQL Injection attempts monthly. The overall cost I incurred in Cloudflare Enterprise was $200 monthly.

3. Case Study: Real-Life Example My media company offers a news portal based on Joomla, where native features and Akeeba Admin Tools were utilized. The native CMS updates decreased the exposure to attacks, while Akeeba Admin Tools offered extra security against XSS attacks and defacements. The cost of the whole plugin was broken down to $50 per year.

6.3. Recommendations for Selecting the Best Security Model

Choosing the right security model is, therefore, paramount and depends on the organization and size of the business. For small businesses or personal websites, the suggestion is that firms focus on using native CMS features for cost and then utilize a low-cost or open-source plugin such as Wordfence or iThemes Security. Although smaller websites are often attacked less frequently, those basic defenses are enough to provide good security with little expense.

It is also recommended that medium-sized organizations stick with native features for basic security and complement them with paid third-party applications. These organizations deal with information that is often classified. They, therefore, have to guard against sophisticated attacks.

Forty-four percent of traffic is enough to rely on third-party tools like enterprise-grade WAFs and monitoring systems for large enterprise-level sites and heavy-traffic web applications that should not neglect native exclusive features of the program, such as RBAC and updates. Larger deployments are more vulnerable to professional threats and require extensive and easily extendible protection.

Those organizations with fewer technical resources should look for tools that are easy to use and where as many functions as possible can be performed automatically, including updates and monitoring from a user-friendly dashboard. This way, there is a comparatively low likelihood of misconfiguration, enabling such simple users to keep sufficient protection strategies.

That is why choosing the native CMS features vs. third-party plugins/tools somewhat boils down to an organization's size/budget/technical proficiency and concrete security needs. Native features allow for establishing a set of primary security requirements, which are sufficient for building elementary protection systems; third-party tools include extra features that enable effective protection against more complex threats. The synergy of the mentioned approaches is the

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

most successful, where one takes care of different levels of vulnerabilities, while the other three are the most profound, balancing the performance, price, and usability. Based on administrators' analysis of the particular organizational requirements and possible resources, choosing the most suitable security model for the CMS platform is feasible.

VII. BEST PRACTICES FOR CMS SECURITY

It is mandatory for any organization that hosts CMS for content management to ensure the security of those platforms. Because CMS platforms are so popular, they are a popular choice for hackers, which means that such platforms can suffer severe consequences, including compromised data and content integrity. Hence, risk management measures should be adapted to include strong security measures. This way, organizations can prevent user data leakage by protecting it with privacy measures and ensuring the integrity of the content while preserving the general organizational assets against threats. The following outlines a broad recommendation to strengthen CMS security and deter risks.

7.1. Guidelines for Securing User Data

The most basic concept when it comes to CMS security is the enforcement of password policies accompanied by MFA. Passwords are the first protection against unauthorized access to a system; therefore, they ensure that all users, especially those with privileged access, use strong and unique passwords. These passwords should be between 12-16 characters consisting of letters in both Upper and lower case, numbers, and symbols. In the same perspective, organizations should not use default credentials and insist on changing passwords as a practice. MFA is just as important as adding a layer of authentication through another authentication factor, such as a code or fingerprint scan. This two-tiered approach greatly minimizes the possibility of illegitimate access and theft of login information.

They also include frequently updating and managing the various available patches relating to a CMS environment. Upgrading the CMS core and any CMS-associated plugins or themes protects from bugs and holes, cutting the risk greatly. Organizations must ensure that it is possible to automatically get updates for minor releases and update and check third-party extensions occasionally, as attackers often breach outdated software. Additionally, getting a hold of security advisories related to the CMS implemented can enable organizations to familiarize themselves with vulnerability notices and fixes. This is because frequently updating an organization's system can help them shut down vulnerabilities and guard their systems from exploitation.

7.2. Strategies to Prevent Content Tampering

Content tampering prevention is also an important aspect of CMS security, and the solution is to use Role-Based Access Control (RBAC). RBAC limits user access to the CMS only by providing certain authorization rights to particular roles rather than giving rights to specific users. This means that users are only allowed to work within the areas specific to their duties, thus strictly controlling what they can do, minimizing the chance of someone modifying the content or settings. For example, in the system, an "Editor" privilege might be created, allowing for the editing of content but not for the administration of the site. Such matters warrant limitations on access to specific accounts to prevent havoc when an account is, for instance, hacked or restrict alterations to be made by everyone but the right people.

Therefore, using HTTPS and a secure database connection is crucial for database security during transmission. Using HTTPS and an SSL/TLS certificate, data transferred between the user and the web server is encrypted, making it impossible for hackers to engage users in a man-in-the-middle attack. Also, incorporating data encryption into database connections means that data is protected before it is transferred to the CMS and the database server. HTTPS must be implemented across an organization's platforms and solutions, mixed content should be checked frequently and systematically, and protection should be in place to ensure that database credentials are well-secured. This all-encompassing approach to securing data transmission adds to the strength of the CMS against extraneous attacks.

7.3 Importance of Regular Backups and Incident Response Plans

The security of CMS cannot fail to have regular backups and well-set incident response policies that need to be in place. Data should also be backed up automatically to include all the CMS files, databases, and configurations if data is lost through a security breach. To reduce the vulnerability of these backups, offsite locations or cloud storage should be used to store the backups; occasionally, the backups should be checked to ensure they are still recoverable. As important as the incident response plan is, it identifies processes for handling security incidents. Ideally, this plan

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

should define the responsibilities of team members and contain recommendations on tracking system logs on the occurrence of suspicious activities. This is because clients are assured of quick recovery and continuity by prioritizing backups and being ready to deal with possible incidents.

Implementing these best practices will go a long way in improving the security of organizations' CMS platforms and reducing the ever-present threats of breaches and data loss. Protection of the users' data begins with using passwords and MFA and updating programs to block unauthorized access. Also, content tampering prevention through RBAC, use of HTTPS, and secure database connection maintains the websites' content and functionality. Last but not least, regular backup and incident response plans prepare organizations for a rapid and deep recovery in cases of cyberattacks.

Everyone within an organization that uses the systems has a role in ensuring that preventive measures are enforced and a culture of security is embraced. Cyber security is not a process to be done but a continuous process that calls for constant monitoring of new threats. Learning the above facts shows that organizations can be assured that their CMS-driven websites serve as reliable information portals following the emergence of advanced threats that compromise entities' reputations to their stakeholders.

VIII. CONCLUSION

This paper focuses on the security challenges of open-source Content Management Systems (CMS) because the internet is a major factor in today's world, and individual freedom and data integrity are under threat. This research has thoroughly aimed at identifying vulnerabilities in open-source CMS platforms, studying practical case scenarios, assessing different security models, and developing the right measures for securing open-source CMS platforms. The lessons learned reveal the need for a more consolidated, coordinated, and comprehensive mental model of CMS security, suggesting that securing such systems cannot be simply an IT issue solely within the domains of IT professionals.

8.1. Recap of Key Findings and Their Significance

Significantly, we found numerous security issues relating to Open-Source Content Management System software. Despite these systems providing high levels of flexibility, they are widely implemented and exhibit very high vulnerabilities because of unpatched software, poor user authentication, and inadequate settings. All the most popular types of attacks, including SQL injection, cross-site scripting, and brute forcing, are potential threats. Failure to address such risks poses dangerous repercussions such as data losses and website hacking, and organizations can suffer severe blows to their reputations.

It also provided information on the practicability of various security measures that have been adopted. Freshening up the software, stringent login procedures, user profiling techniques, and the combination of security plugins and tools can prevent cyberattacks completely. For instance, firewalls and updating software were among the most effective measures to improve security. Moreover, the need to compare native security features of CMS platforms with extensions was identified; although native features may suffice, there are additional solutions with stronger, tweakable security. However, care must be taken not to add other forms of vulnerability, as the tools ought to be used to supplement the existing mechanisms.

The third and rather important lesson that is evident in the research is the concept of the trade-off between use and use! Proposed security policies may make users uncomfortable and cause them to opt out of using the platforms, while lax security policies may expose the platforms to malicious attacks. Adjusting risk exposure adequately is critical to achieve the right level of security while leaving the system as open as possible so as not to be inconvenient.

8.2. Final Recommendations for Improving CMS Security in Open-Source Platforms

A more systemic approach is advised to secure CMS better. This involves the persons integrating a combination of technical sanitation and administrative rules. To enhance the security of their CMS, organizations should implement policies like a strong password policy and MFA and ensure software, plugins, and themes are updated. Furthermore, the protection from the most frequent threats can be enhanced by using trusted third-party security plugins or services, such as Wordfence or Sucuri.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Adopting Role-Based Access Control (RBAC) and enforcing it are the other major recommendations. To reduce potential changes, which can be either inadvertent or intentional, organizations must restrict the ability of users to make changes by assigning them roles and permissions. The periodic review of all users of different roles will assist in displaying any personnel who would wish to have access to the critical features of the website. Also, data transmission over HTTPS and using a secure connection string for the database to manage data are sensitive factors during its transfer.

This is important since it helps one know the areas that can be exploited if not taken care of during security checks. An organization can design and implement tools for vulnerability assessments, including the OWASP ZAP or Burp Suite. Security training involving developers, administrators, and ordinary users prevents human factor-induced problems, which constitute some of the most common security issues.

Organizations should also ensure that they back up their data; in case of an attack, this data would act as a last resort to the attack. Point-in-time copies with encrypted backup will make it possible to rerun from the previous backup as soon as possible with little or no interruptions. Finally, creating the incident response plan will help organizations prepare for a security breach; it will contain procedures for identifying and eradicating threats and methods for informing the affected parties.

8.3. Call to Action for Developers and Administrators

Security is everyone's duty, so CMS security doesn't lie solely with the developers, administrators, or end-users but with everyone involved in the system. Open-source creators need to routinely enhance security to prevent hackers' unauthorized access and compromise organizations using such platforms. System administrators should be more active, revise many settings regularly, check for abuses, and regularly set up security parameters.

Another stakeholder group in this process is the open-source community, which promotes collaboration by sharing knowledge, tools, and examples of good practice. This way, developers and administrators can reduce the security risks of known CMS platforms, allowing users maximum protection and website content integrity.

In conclusion, the security of open-source CMS platforms must be regarded as a fundamental priority rather than an afterthought. With countless websites relying on these systems, fortifying their defenses against evolving threats is essential. By implementing the recommended practices and embracing a security-first mindset, organizations can safeguard their digital assets and contribute to a more secure online ecosystem. The time to act is now—secure your CMS platform today and contribute to building a safer future for the web.

REFERENCES

[1] Meike, Michael & Sametinger, Johannes & Wiesauer, Andreas. (2009). Security in Open Source Web Content Management Systems. Security & Privacy, IEEE. 7. 44 - 51. 10.1109/MSP.2009.104.
[2] Zamościński, Patryk & Koziel, Grzegorz. (2020). Analysis of security CMS platforms by vulnerability scanners. Journal of Computer Sciences Institute. 16. 261-268. 10.35784/jcsi.2020.
[3] Petkova, Lilyana & Pavlova, Vasilisa. (2022). Security Analysis on Content Management Systems. Mathematics and Informatics. LXV. 423-434. 10.53656/math2022-5-2-sec.
[4] Ziakis, Christos & Vlachopoulou, Maro. (2021). Web Content Management Systems used by Search Engine Optimization Experts for Top Rankings in Search Engine Result Pages. WSEAS TRANSACTIONS ON COMPUTERS. 20. 207-216. 10.37394/23205.2021.20.22.
[5] Ismailova, Rita. (2017). Web site accessibility, usability and security: a survey of government web sites in Kyrgyz Republic. Universal Access in the Information Society. 16. 257-264. 10.1007/s10209-015-0446-8.
[6] Asaduzzaman, Md & Rawshan, Proteeti & Liya, Nurun & Islam, Muhammad & Dutta, Nishith. (2020). A Vulnerability Detection Framework for CMS Using Port Scanning Technique.
[7] Kk, Madhura. (2024). APPLICATION OF AI AND BLOCKCHAIN TECHNOLOGY IN DCMS FOR THE AUTOMATIC DOCUMENT CLASSIFICATION AND IMPROVE THE SECURITY. 10.58532/V3BFCT3P3CH5.
[8] Yaseen, Kamal Aldin & Yaseen, Yousif. (2023). JOURNAL OF CRITICAL REVIEWS Enhance MOODLE Platform Security Against Denial of Service Attack (DoS). 10. 140-147. 10.31838/jcr.10.02.16.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

[9] Almuhtadi, Wahab & Fenwick, Wynn & Henley-Vachon, Liam & Mitchell, Peter. (2022). Assessing Network Infrastructure-as-Code Security using Open Source Software Analysis Techniques Applied to BGP/BIRD. 1-6. 10.1109/ICCE53296.2022.9730211.

[10] Kavithamani, C. & Subramanian, R. & Krishnamurthy, Srinevasan & Chathu, Jayakrishnan & Iyer, Gayatri. (2021). An Analysis of Remotely Triggered Malware Exploits in Content Management System-Based Web Applications. 10.1007/978-981-15-5285-4_15.

[11] [11] G. McGraw, "Software Security: Building Security In", Addison-Wesley, Boston (2006). [12] G. Hoglund, G. McGraw, "Exploiting Software: How To Break Code", Addison-Wesley, Boston (2004).

[13] Symantec Internet Security Threat Report, Trends for January-June 07, Volume XII, http://www.symantec.com/threatreport/ (2007).

[14] E. Jonsson, "Towards an Integrated Conceptual Model of Security and Dependability", Proceedings of the 1st Intl. Conference on Availability, Reliabil-ity and Security (ARES'06), pp. 646-653 (2006).

[15] M. Howard; D. LeBlanc, "Writing Secure Code", Microsoft Press, ISBN 978-0735615885 (2001).

[16] R. Newman, "Cybercrime, Identify Theft, and Fraud: Practicing Safe Internet – Network Security Threats and Vulnerabilities", Proceedings of the 3rd Annual Conference on Information Security Cur-riculum Development, pp. 68-77 (2006).

[17] A. Tanenbaum, M. van Steen, "Distributed Systems – Principles and Paradigms", Prentice Hall Interna-tional, New York (2002). [18] Drupal 6.0 released, http://drupal.org/drupal-6.0 (2008

[18] M. Meike, J. Sametinger, A. Wiesauer, Security in Open Source Web Content Management Systems, IEEE Security and Privacy Magazine, 2009.

[19] S.K. Patel, V.R Rathod, S. Parikh, Comparative Analysis Of Web Security In Open Source Content Management System, ISSP, 2013.

[20] S.K. Patel, V.R Rathod, S. Parikh, Joomla. Drupal and WordPress - A Statistical Comparison of Open Source CMS, IEEE, 2011

[21] Rele, M., & Patil, D. (2023, September). Machine Learning based Brain Tumor Detection using Transfer Learning. In 2023 International Conference on Artificial Intelligence Science and Applications in Industry and Society (CAISAIS) (pp. 1-6). IEEE.

[22] Chandrashekar, K., & Jangampet, V. D. (2020). RISK-BASED ALERTING IN SIEM ENTERPRISE SECURITY: ENHANCING ATTACK SCENARIO MONITORING THROUGH ADAPTIVE RISK SCORING. INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING AND TECHNOLOGY (IJCET), 11(2), 75-85.

[23] Chandrashekar, K., & Jangampet, V. D. (2019). HONEYPOTS AS A PROACTIVE DEFENSE: A COMPARATIVE ANALYSIS WITH TRADITIONAL ANOMALY DETECTION IN MODERN CYBERSECURITY. INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING AND TECHNOLOGY (IJCET), 10(5), 211-221.

[24] Eemani, A. A Comprehensive Review on Network Security Tools. Journal of Advances in Science and Technology, 11.

[25] Eemani, A. (2019). Network Optimization and Evolution to Bigdata Analytics Techniques. International Journal of Innovative Research in Science, Engineering and Technology, 8(1).

[26] Eemani, A. (2018). Future Trends, Current Developments in Network Security and Need for Key Management in Cloud. International Journal of Innovative Research in Computer and Communication Engineering, 6(10).

[27] Eemani, A. (2019). A Study on The Usage of Deep Learning in Artificial Intelligence and Big Data. International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), 5(6).

[28] Nagelli, A., & Yadav, N. K. Efficiency Unveiled: Comparative Analysis of Load Balancing Algorithms in Cloud Environments. International Journal of Information Technology and Management, 18(2).

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details